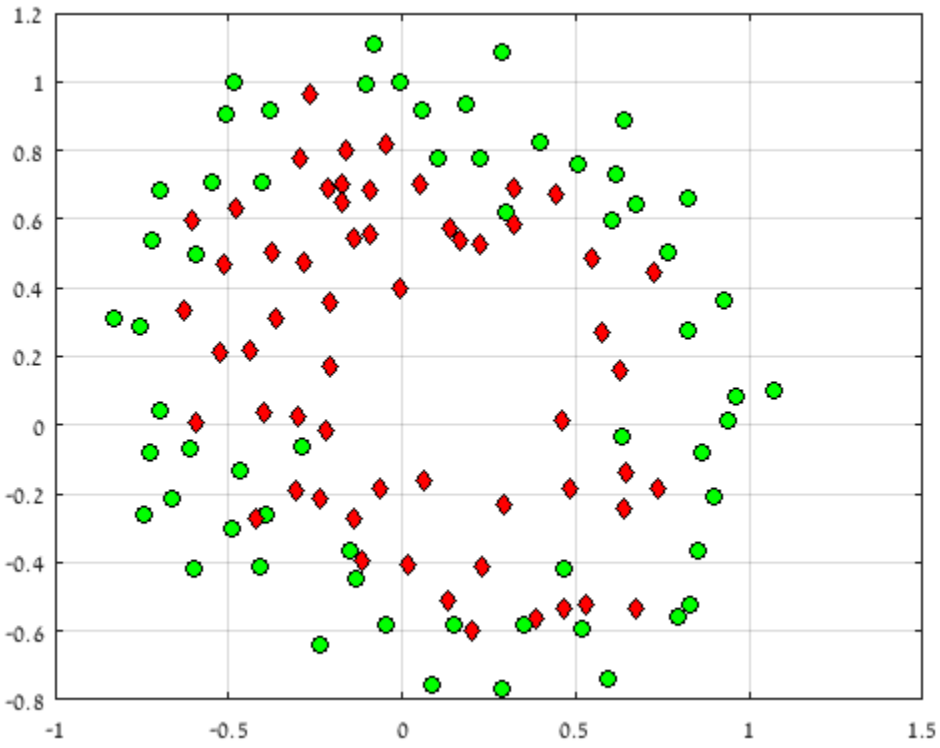
Table of Contents

Ex - 2:

```
clear;
clc;
close all;
data = load('email_data.txt');
```

Section a

```
X = data(:, [1, 2]);
y = data(:, 3);
plotdata(X,y);%plot data each y val with diffrent color
hold on;
```



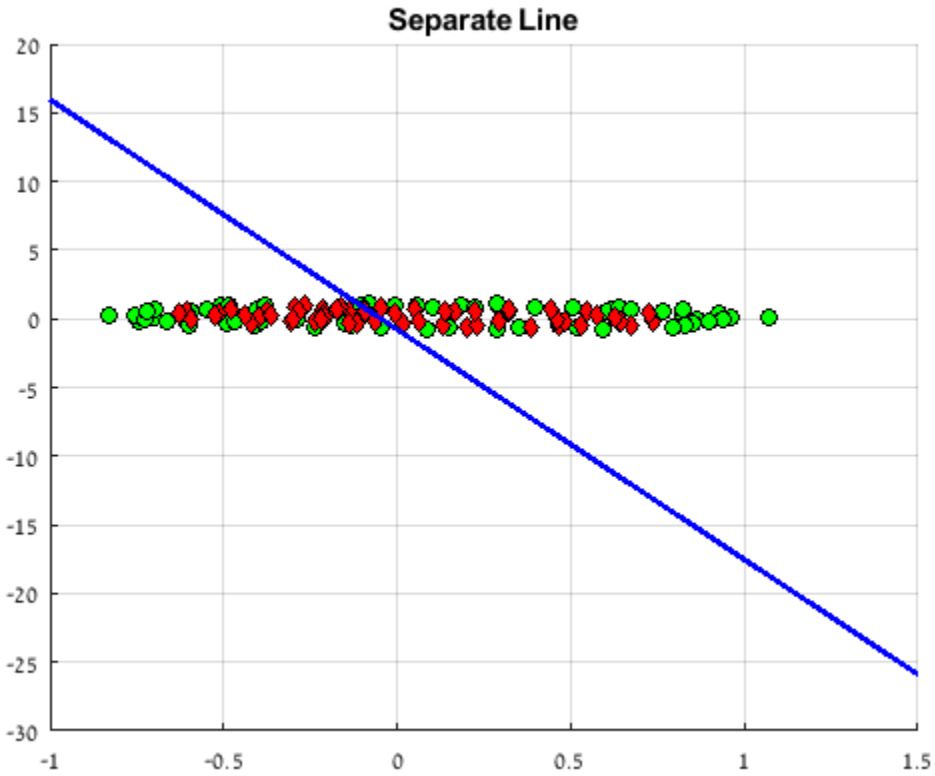
Section b

```
alpha=0.1;
X1=[ones(length(y),1) X]; % concatinate the first column of ones
theta=zeros(size(X1,2),1);

numOfIterations=10000;
[theta,J]=gd(X1,y,theta,alpha,numOfIterations);
plotLine(X1,y,theta); %printing line using the function we create at
 the class


fprintf('My conclusion is that logistic regresion is not enough to');
fprintf(' separate the values\n');
```

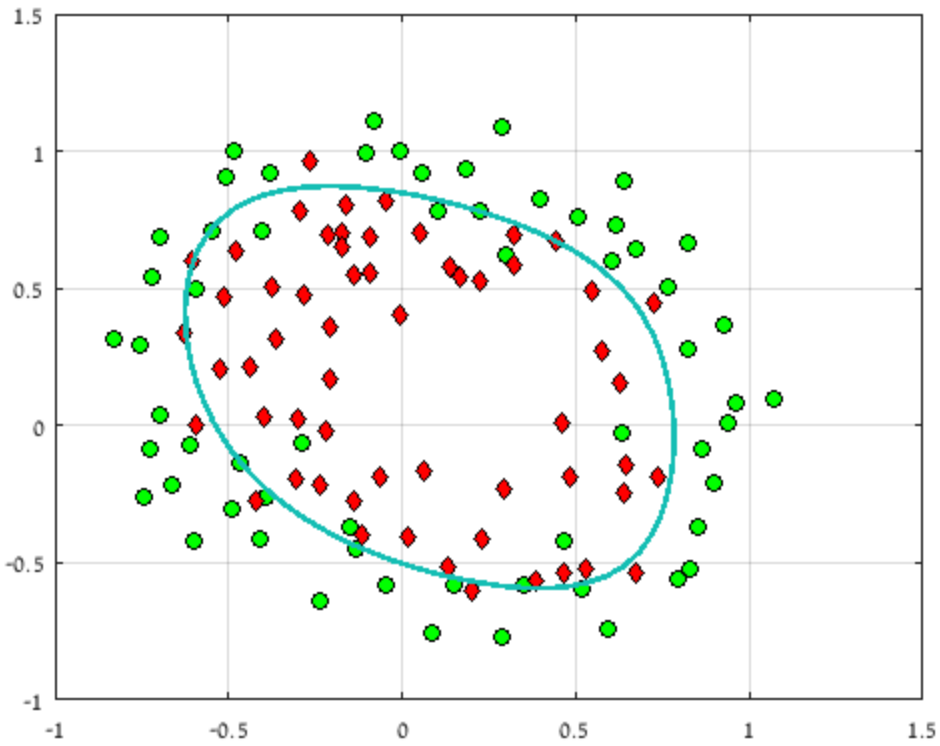
My conclusion is that logistic regresion is not enough to separate the values



Section c + d

```
X2 = mapFeature(X(:,1),X(:,2));
theta2 = zeros(size(X2,2),1 );
lambda0 = 0;
alpha = 0.1;
[theta2,J] = gd_reg(X2,y,theta2,alpha,numOfIterations,lambda0);
```

```
plotDecisionBoundary(theta2,X2,y);
grid on;
```



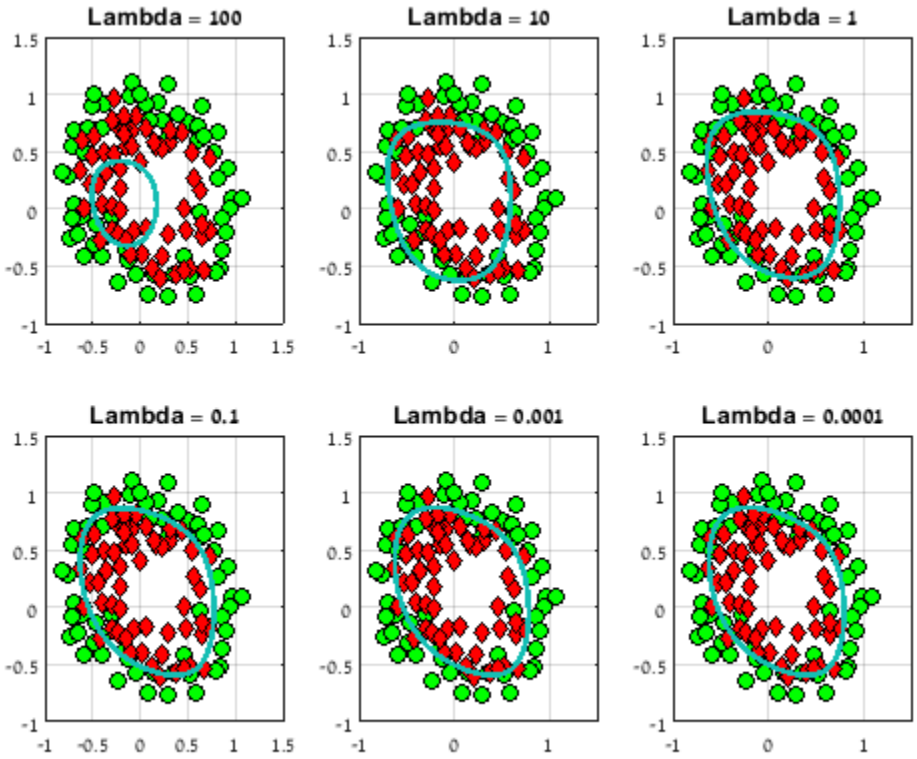
Section e

```
lambda = [100 10 1 0.1 0.001 0.0001 ];
%trying lambda values to see if there is a diffrent between each of
 them
for i=1:length(lambda)
    subplot(2,3,i);
    theta3 = zeros(size(X2,2),1 );
    [theta3,J1] = gd_reg(X2,y,theta3,alpha,numOfIterations,lambda(i));
    plotDecisionBoundary(theta3,X2,y);
    title(sprintf('Lambda = %g',lambda(i)));
    grid on;
end

fprintf('We can see that if the lambda values are higher then 1,\nthe
 circle');
fprintf(' is getting smaller, and if the lambda values are smaller
 then 1\n');
fprintf('the circle stay almost (for my eye) the same and can classify
 better.\n');
```

We can see that if the lambda values are higher then 1,

the circle is getting smaller, and if the lambda values are smaller then 1
the circle stay almost (for my eye) the same and can classify better.



Section f

```
data2 = load('emaildata3.mat');

dataX = data2.X;
dataY = data2.y;
XX = mapFeature(dataX(:,1),dataX(:,2));

predictAndPrintTheError(XX,dataY,theta2)
```

The error prediction is on 0.09 precent

Section g

```
title('Repeat section c+d');hold on;

theta0=zeros(size(XX,2),1);

%the optimal
close all;
options = optimset('GradObj', 'on', 'MaxIter',numOfIterations);
```

```
title('For lambda = 0:');hold on;
[theta4, J2, exit_flag] = fminunc(@(theta0)(cost_log_reg(theta0, XX,
 dataY, lambda0)), theta0, options);
plotDecisionBoundary(theta4, XX, dataY)
grid on;

pause(3);

fprintf('lets see the diffrent between the prediction of the gd and
 the fminunc:\n');

fprintf('For gd: ');
predictAndPrintTheError(XX,dataY,theta2)

fprintf('For fminunc: ');
predictAndPrintTheError(XX,dataY,theta4)

fprintf('We can learn from this that the fminunc predict better then
 the gd.\n');

lambda = [0 0.1 0.01 0.0001  0.00001 0.00000000001];
%testing diffrent lambda values
title('Repeat section e');hold on;

for i=1:length(lambda)
    subplot(2,3,i);
    %try it now with diffrent options
    [theta5, J3, exit_flag] = fminunc(@(theta0)(cost_log_reg(theta0,
 XX, dataY, lambda(i))), theta0, options);
    plotDecisionBoundary(theta5, XX, dataY)
    title(sprintf('Lambda = %g',lambda(i)));
    fprintf('For lambda %g: ',lambda(i));
    predictAndPrintTheError(XX,dataY,theta5)
    grid on;
end
%we can see the diffrents between higher lambda values and lower then
 0.00001
%lambda values
%I learned from it that if the lambda value are closer to 0 or is a 0,
%the fminunc predict in 99% meaning the error is in 1%.
%if the lambda is not exactly zero or closer to it, the prediction
%getting worst.
```

```
Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.



lets see the diffrent between the prediction of the gd and the
 fminunc:
For gd: The error prediction is on 0.09 precent
```

For fminunc: The error prediction is on 0.01 precent
We can learn from this that the fminunc predict better then the gd.

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 0: The error prediction is on 0.01 precent

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 0.1: The error prediction is on 0.28 precent

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 0.01: The error prediction is on 0.28 precent

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 0.0001: The error prediction is on 0.11 precent

Local minimum possible.

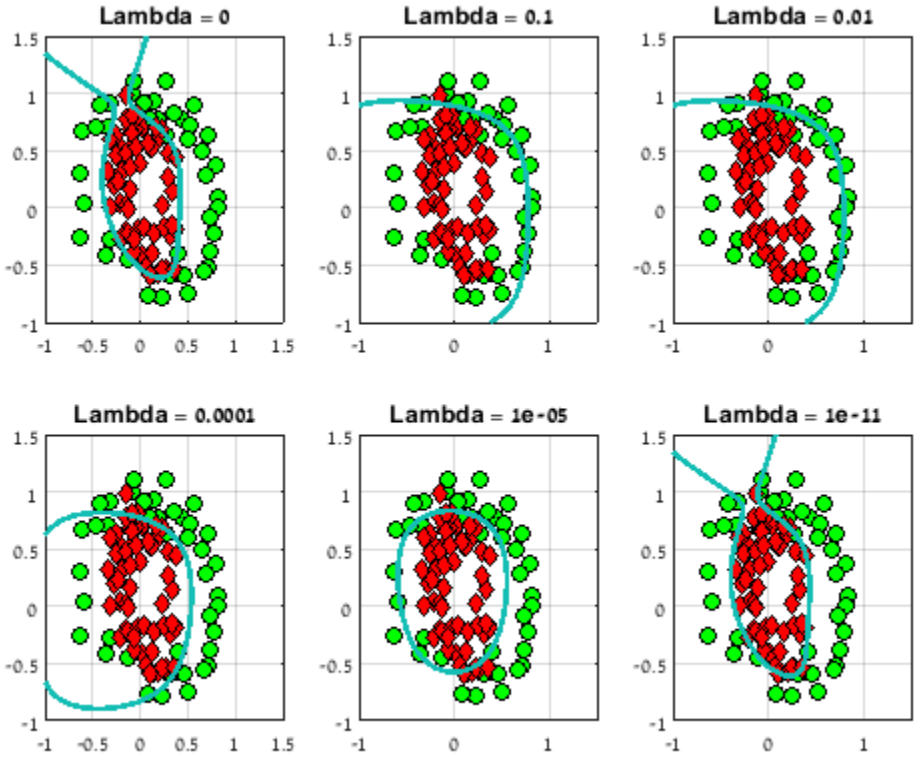
fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 1e-05: The error prediction is on 0.01 precent

Local minimum possible.

fminunc stopped because it cannot decrease the objective function
along the current search direction.

For lambda 1e-11: The error prediction is on 0.01 precent



Published with MATLAB® R2018a